視覚情報を用いた自律走行車のナビゲーション* (情報の性質により分離した階層化コントローラの実験的検証)

植村 義彦^{*1}, 中澤 和夫^{*2}

Navigation of intelligent vehicle using visual information (Experimental Evaluation of Hierarchical controller Divided by Information Type)

Yoshihiko UEMURA and Kazuo NAKAZAWA

We aim at developing an intelligent machine by taking up an intelligent vehicle that is to get to a destination without touching walls on corridors in a building using visual information. Robustness of systems and facility of designing are considered important on the ground that it is indispensable to achieve intelligent machines in real world. Therefore, we adopt a hierarchical architecture which has a NAV layer described as mode representation including map information and DRV layer made up of some behavior-based-modules. Moreover, interpreting visual information qualitatively avoiding excessively quantitative interpretation to recognize characteristic places allows the system to get to the destination without artificial landmarks. An experiment with a vehicle which is fully equipped with camera devices, computing devices, battery, and driving system showed the effectiveness of this method.

Key Words: Moving Robot, Intelligent Vehicle, Optical Flow, Autonomous Navigation, Mode Transfer Representation, Behavior-Based

1. 緒 言

近年のメカトロニクス技術やコンピュータ技術の発展に伴い、自動車の経路制御を自律化する研究⁽¹⁾が多くなってきている。また実験レベルではあるが ITS⁽²⁾などにおける無人走行実験など自動走行させる研究が盛んである。しかしそれらはガイドライン、磁気ネール⁽³⁾⁽⁴⁾などのインフラを整備する必要があるため、広い範囲で実現しようとするコストの点で問題となる。したがってこれらの問題を克服するためには、受動的手法により得られるセンサ情報のみを用いて走行車を誘導する手法の開発が望まれる。

筆者らは、既に視覚誘導技術の確立のために廊下の壁に衝突しないように実環境・実時間にて走行する自律走行車⁽³⁾を開発してきた。これは、CCDカメラより得られる周囲の画像からオブティカルフローを計算し、壁に衝突しないようにステアリング角の制御をするものである。この走行車は、直線の一般的な建物内の廊下にて秒速1000mm/Sで自律的に壁との衝突を回避しながら移動することができる。これは、基本的には Reactive Behavior に基づいたコンセプト⁽⁴⁾と同様で

ある。この考え方の特徴は、単調な環境に対しては十分に安定した走行を実現する事ができることである。しかし複雑な経路を走行させるためには、センサと行動を密接に関連した低レベルな行動の制御のみでは、コストや時間がかかってしまう。そこで低レベルな環境認識・行動モジュールと、状態遷移図によって記述された誘導計画モジュールに分け、効率良く誘導計画を行う手法の提案をする。環境認識・行動モジュールはセンサやアクチュエータに直接対応した情報を取り扱いその周期も短い。これに対して誘導計画モジュールは、ナビゲーションのプランニングを行うといった記号レベルの情報を取り扱い、その処理の実時間性の要求度は低い。したがって情報の性質により処理するコントローラを階層化することにより効率の良いプログラミングが期待できる。

本論文では、以下提案する概念について述べ、さらに1/10モデルの走行車を用いた実走行により、本手法の有効性と問題点を明確にする。

2. 誘導計画のアーキテクチャ

走行車を視覚情報を用いて誘導するためには、次の2つのレベルの機能が要求される。

1) 始点から終点までの経路を何らかの方法で記述した地図にしたがって誘導するナビゲータ。

* 原稿受付 1997年5月27日。

^{*1} 慶應義塾大学大学院 [現: 有人宇宙システム(株)]。

^{*2} 正員, 慶應義塾大学理工学部 (☎ 223-8522 横浜市港北区日吉3-14-1)。

Fig.1 Architecture of the system

2) 現在走行している道路で安定した走行を担当するドライバ。

ナビゲータは道路状況の大まかな特徴から経路全体の中での位置を判断するものである。ドライバから周囲の状況を聞きながら、直線路、十字路、L字路などを認識し、進むべき経路の判断を行い、その結果をドライバーに伝える。

ドライバは自分の視覚から周囲の状況を判断し、現在走行中の道路内で安定に走行することに専念する。また見えた状況をナビゲータに伝え、経路上の位置を聞きステアリングの操作を行う。以上の作業を行うために提案するシステムの構成を図1に示す。NAVはナビゲータ、DRVはドライバを示し、2つに分割することにより機能を分担する。まずDRVでは視覚センサから得られる情報から走行車の走行状態を認識する。具体的には廊下のドアや配電盤、消化器などから生成されるオプティカル・フローを算出し⁽⁵⁾、現在走行車が直線状の道路の中央を走行しているのか、あるいは端を走行しているのか。また右に回転しているのか、左に回転しているのかなどの状況を認識する。認識結果はNAVに伝える一方でステアリング、アクセル、ブレーキの操作を行う。NAVではDRVから送られてくる状況判断に応じて現在位置の確認を行う。ここでは状態遷移図を用いて誘導の記述を行った。DRVより得られる視覚認識結果から現在の状況を遷移させていきその結果をDRVに伝えるという作業を行う。

以下、各層の役割について述べる。

2.1 環境認識・行動モジュール層 オプティカルフローを入力として反射的な行動を行う行動型 (Behavior-based) モジュール群を図1のようにDRVに配置する。本システムの視覚情報は1次元CCDカメラより得られる1次元輝度画像である。視覚情報を1次元とすることで低コストで実時間処理、分解能の向上、ダイナミックレンジの向上が実現できる。このカメラを用いて環境情報を取得し、オプティカルフローの算出を行うことにより周囲の空間的な状態を把握することができる。

まずオプティカルフローから空間の把握法について述べる。図2はカメラの空間的位置関係を示している。カメラのレンズ中心を原点としたカメラ座標系を定義する。 $x-y$ 平面上にある特徴点 $P(x_p, y_p)$ の像点 Q の座標 u_q は、焦点距離を f とすると、図2に示すように幾何学的関係から次式のように算出できる。

$$u_q = f \frac{x_p}{y_p} \quad (1)$$

ここでは自走車はある一定速度で直進しているものとする。すると P 点の移動先の点が P' 、その像点が Q' となる。以上の位置関係から、特徴点 P のカメラ座標系における位置は、

$$x = \frac{u_q u_{q'} V t}{f(u_{q'} - u_q)} \cdot w \quad (2)$$

$$y = \frac{u_q V t}{u_{q'} - u_q} \quad (3)$$

となる。ただし w はCCDカメラの1画素の長さである、 V は走行速度、 t は走行時間である。これらの特徴点の位置的分布図から環境を判断する。

次に得られた結果からステアリング角を決定する方法について述べる。まずDRV層の各モジュールのうち

Fig.2 Optical system

直線廊下走行モジュール(Straight)を例に説明する. 先に算出された特徴点の分布を中心画素を境に左右に分割する. そしてそれぞれの特徴点について x 座標の平均値を算出し, 次式によりその偏差を算出する.

$$err = \frac{\sum_{leftnum} x_{left}[i]}{leftnum} - \frac{\sum_{rightnum} x_{right}[i]}{rightnum} \quad (4)$$

そして次式によりステアリング角を算出する.

$$steer = k \cdot err \quad (5)$$

ここで k は操舵の安定性を決定する係数でシミュレーションと実験により決定する. 壁に対して平行に近い角度であれば, 位置情報のみにより安定に走行できる.

同様に旋回動作では一定方向にオプティカルフローが生じるようにステアリング角を決定する.

2.2 誘導計画モジュール層 状態遷移図は自走車の現在の状態を表す状態と, その間の遷移関係の情報を持つ. 各状態は遷移条件の情報を保持しておりスタートから始まり環境情報の入力条件を満たすとそれに従い状態を遷移させていく. 十字路(Cross-road)や直線廊下(straight-corridor)などを走行していることが状態に相当する. 状態遷移のための環境情報は, 実世界のノイズ, 処理の誤りなどを考慮する必要がある. したがって視覚情報を過度に定量的に解釈せずに, その環境固有の特徴量を定性的に判別した. つまり自律走行車の走行位置などの状態の変化に対しても安定に判別できることが重要である. 環境認識部からはオプティカルフローより求められた周囲の特徴点の2次元的な分布を得ることができる. したがってこの情報から環境を定性的に表現すると以下ようになる. 直線廊下においてはその幅は一定であるため, 左右の壁毎の特徴点の x 軸方向の標準偏差は少ない. 十字路にては廊下の幅はある程度広く, 左右の壁の特徴点分布は大きく認識され, L字路は左曲がりであると

Fig.3 Probability density function

Table 1 Shape parameter of Probability density function

(1) Straight corridor

$i(x[i])$	ft1	st1	st2	ft2	Wi
0	0	200	500	900	5
1	0	200	500	900	5
2	1500	2000	4000	5000	0
3	1500	2000	4000	5000	0
4	1000	2000	2600	3000	3
5	900	1400	2000	3000	3
6	0	3	6	9	2
7	0	3	6	9	2

(2) L type corner

$i(x[i])$	ft1	st1	st2	ft2	Wi
0	700	1000	2500	4000	5
1	400	600	1500	3000	5
2	4000	5000	9000	12000	2
3	0	1000	3000	4000	2
4	2000	2500	5000	7000	3
5	1300	2000	4000	7000	3
6	2	5	8	15	3
7	2	5	8	15	3

(3) Cross road

$i(x[i])$	ft1	st1	st2	ft2	Wi
0	0	500	1000	1500	5
1	500	1000	1500	2500	5
2	2000	4000	7000	9000	0
3	0	1000	1500	3000	0
4	2000	3000	6000	7000	3
5	1700	2500	4000	6000	3
6	4	8	15	20	2
7	4	8	15	20	2

Table 2 Threshold of Mode transfer

Corridor 1

Th_cur	0.4
Th_next	0.8

Corridor 2

Th_cur	0.2
Th_next	0.7

すれば右側の壁の標準偏差が小さくなる.

そこでこれらの特徴量を入力とする各環境, 各特徴量に対し図3のような確率密度関数を定義する. これらを各特徴量の重要度に応じて重みづけを行い正規化しその環境らしさを表す環境の指標とする.

$$Idx = \frac{\sum_i w[i] \cdot \sum_j f_j(x[i])}{\sum_i w[i]} \quad (6)$$

これらは各環境毎に定義されるものであり, 0~1の値を持ち, 1に近いほどその環境らしいということになる. 表1に各環境における確率密度関数の形状パラメータと重みを示す. 実際に(1)直線廊下, (2)十字路, (3)L字路を走行し, 各々の環境のパラメータを経験的に決定した. また(3)のL字路は, 左折用であるが, 右折用の形状パラメータは同一形状のL字路であれば左

Fig.4 Mode transfer representation

右のパラメータを入れ替えば良い. 表中の縦項目の0と1はx軸方向の左右の特徴点の分散, 2と3は同様にy軸方向, 4はx軸方向の全特徴点の分散, 5は廊下の幅, 6は左壁上の特徴点の数, 7は右壁上の特徴点の数を示す.

次に状態の遷移法について述べる. 現在の環境(状態)の指標と遷移先の環境(状態)の指標のそれぞれのしきい値を越えたとき状態を遷移させる. つまり式(6)(7)の条件式を満たせば次の状態に遷移する.

$$\text{Idx}(\text{current}) < \text{Th}_{\text{cur}} \quad (7)$$

$$\text{Idx}(\text{next}) > \text{Th}_{\text{next}} \quad (8)$$

これらのしきい値も実走行により得られた結果から求めたものであるが, その数値を表2に示す.

図4はある環境を走行したときの状態の移り変わりを示している. 各状態はDRV層の各々行動に基づくモ

ジュールへ対応しており, 例えば直線廊下という状態においては直進モジュール, 十字路においては左折モジュールというように対応している行動モジュールの出力を有効にさせ, 他のモジュールの出力を抑制することにより, システムとしての出力を得る.

3. システム構成

自律走行車の全体構成を図5に示す. 自律走行車の車体としては1/10スケールの市販のラジオコントロールカーをベースにした. この車体に, CCDカメラ, CPUボード(2枚), I/Oボード, 電源用電池を搭載して完全な自立型システムとした. CPUとしては25MHzのトランスピュータT805を用いた. またI/OボードはA/Dコンバータ, D/Aコンバータ, CCD入力駆動パルス発生回路, CCD出力信号増幅回路, サーボモータ制御回路から構成される. CCDリニアセンサは1024画素のTCD132Dを用いる. CCDのシフトパルスで同期

Fig.5 Hardware system of the Intelligent Vehicle

を取り、出力信号を A/D 変換して CPU ボード上のメモリに取り込む。CCDの同期信号の周期は10mSに設定した。これでオプティカルフローの算出に約数十mSを要する。操舵角は、D/A コンバータを通してステアリングサーボモータおよび走行用モータを制御する。DRV 層と NAV 層はそれぞれのトランスピュータボードに配置し、シリアルリンクを介して状態情報のやり取りを行っている。

制御プログラムは Parallel-C Ver2.2.2(スリーエル社製)により記述する。プログラム開発に用いるノートパソコンにはPCM/CIAカードからトランスピュータへのインターフェースボードが接続されており、これらを通してプログラムをロードする。プログラム実行開始時はインターフェースボードと自走車上の CPU ボードのリンクを接続しておき、実行後はこの接続を切ることによって車両は自立的に走行することができる。

Fig. 6 Map of experiment

Fig. 7 Demonstration of navigation

4. 走行実験による検証

走行実験は、図6に示すように、十字路、L字路、直線廊下が存在する環境中で行った。予め予備実験を行い各状態における環境の指標のための各パラメータを経験的に獲得しておく。

実験環境における直線廊下、十字路、L字路における自律走行車の様子が図7である。図中のaからiは図6のそれに対応している。自律走行車は直線廊下においては特に走行に失敗することなく安定して走行した。十字路やL字路の認識においてまれに認識を失敗することがあるがほぼ安定して目的地に到達することができた。

表3に数回行った走行実験の結果を示す。表中のSは成功、Fは失敗を表す。この表に示されるとおり、自律走行車は低レベル走行における直線廊下の直進では100%壁への衝突を回避しており、かなり安定して走行をおこなうことができると言える。失敗の多くはNAV層における状態の認識の失敗に起因するものであると考えられる。例えば、十字路で若干認識の不成功がある。全く十字路を認識しなかったのは13回中1回にとどまった。残りの2回は認識をしたが若干早すぎて、あるいは遅すぎて十字路における行動Turn rightで壁に衝突したものである。L字路においては十字路における認識よりも高い確率で認識を行うことができ

る。これは、L字路においては自律走行車の正面に壁があり、直線廊下と区別が付きやすいことが考えられる。

5. 結 論

以上本論文では、1次元輝度画像より移動ステレオ法に基づいて環境中に存在する特徴点の位置分布を算出し、これから抽出した情報を用いて自律走行車を目的地にまで到達させる手法を開発し、その有効性を確認した。今後の課題としてはシステムの安定性の向上、状態認識のための各パラメータを自動的に獲得をさせることなどが挙げられる。

文 献

- (1) 高橋, 複数の走行環境情報を扱う自動車制御のための情報融合に関する研究, 機論, 63-608, C (1997), pp. 146-153
- (2) ITS 日米欧3極比較レポートPARTII, (1997).
- (3) 上田敏, 木村邦久, 村田重雄, 自動運転道路システムの開発, 電気学会研究会資料, RTA-96-13, (1996), pp. 1-10.
- (4) 小川寛一, 我が国における次世代道路交通システムの動向と展望, 電気学会研究会資料, RTA-97-19, (1997), pp. 47-55.
- (5) 山田他, オプティカルフローによる自走車のビジュアルナビゲーション, ロボティクス・メカトロニクス95講演会論文集, (1995), pp. 485-488.
- (6) R. A. Brooks, A Robust Layered Control System for a Mobile Robot, IEEE Journal of Robotics and Automation, Vol. 2, No. 1, (1986), 14-23.
- (7) A. Singh, Optical Flow Computation, IEEE Computer Society Press, (1991).

Table 3 Results of navigation

	Total	Straight1	Cross	Straight2	L-shape	Straight3	Goal	Reason of Failure
1	S	S	S	S	S	S	S	
2	F	S	S	S	F			Unexpected dynamic obstacles
3	F	S	F					Early recognition at Cross
4	S	S	S	S	S	S	S	
5	S	S	S	S	S	S	S	
6	F	S	F	S	S	S	S	Late recognition at Cross
7	F	S	F					No recognition at Cross
8	S	S	S	S	S	S	S	
9	S	S	S	S	S	S	S	
10	F	S	S	S	F			No recognition at L-shaped
11	S	S	S	S	S	S	S	
12	F	S	F	S	S	S	S	Late recognition at Cross
13	S	S	S	S	S	S	S	
%	66.7	100	69.2	100	90.0	100	100	